

IV. REMARKS

1. The title is amended.

2. Claims 1-4, 9-12, 15, 18, 21 and 22 are not anticipated by Bayeh et al. ("Bayeh") U.S. Patent No. 6,098,093 under 35 U.S.C. §102(e).

Claim 1 is directed to managing a plurality of sessions. The sessions are between a plurality of terminals and a server. The server has a plurality of threads. The sessions are grouped into a plurality of groups and a "thread" is assigned to each "group" of sessions. This is not disclosed or suggested by Bayeh.

In Bayeh there is no control mechanism for a thread to handle the requests of a particular session. Instead, any server can handle the requests of a particular session and a mechanism is present to provide the session object to that server. There is no **assignment of a thread to a session** because sessions are not tied to any particular server. Furthermore, because there is no assignment of a thread to a session, there cannot be assignment of a thread to a **group of sessions**. Therefore, claim 1 cannot be anticipated by Bayeh.

The concepts underlying the claimed features of Applicant's invention are quite different from Bayeh. Bayeh relates to a technique, system, and computer program for maintaining session-related state information in a scalable, clustered network environment (column 1, lines 8 to 10). Bayeh is concerned with providing the necessary session information (state information) to enable session requests to be handled. Sessions are identified by session IDs and state information, describing the state of a session, is stored on the server using a session

object. This object is created when a new client session begins, and is kept for the duration of the session. The object stores information about the transactions occurring between the client and the server (column 4, lines 5 to 9) and is shared among servlets (column 4, line 23). In terms of organizing sessions, Bayeh is explicit in stating that [w]hen session services are provided at these clustered servers, the group of sessions can logically be thought of as a *pool, or cluster, of sessions*. Since the *Session Tracking facility in the Java Web Server Toolkit is only valid within the scope of a single Web server*, the *pool of sessions cannot be properly maintained among a group of clustered Web servers such as this*.

For example, if a client request is received at one server, and that server maintains information about the on-going session, there is no way for this version of the session information to be made available to a different server in the cluster if the next request from this client goes to a different server. (Column 4, line 59 to column 5, line 3.)

Thus, Bayeh teaches that sessions are identified by session IDs, session state information is maintained as session objects, the objects are shared among servlets which handle session requests, and that in a clustered server environment, one server might handle a request for a particular session and then another server might handle a subsequent request for that session. In other words, the particular session is not associated with a particular server because different servers might handle the requests for that session.

Bayeh is concerned with the discovery and retrieval of the appropriate session object in handling a request. Because Bayeh is a clustered server environment, when a request arrives which

is part of a session, since the environment is not arranged to route the request to a particular server to deal with the request, the server receiving the request needs to be provided with the session object which is relevant to that session. Therefore, [t]he servlet engine configured as a session server further comprises a subprocess for maintaining a plurality of session objects; a subprocess for locating one of the session objects in response to a request from one of the servlets or one of the session clients; and a subprocess for returning the located session object in response to the request (column 5, line 65 to column 6, line 4). This is not the same as in Applicant's invention, where the sessions are grouped into a plurality of groups and a "thread" is assigned to each "group" of sessions, and the assigned thread only handles the events of the group of sessions.

Also in Bayeh, a load-balancing host 59 functions as a type of front-end processor to these servers, receiving client requests 100, 101, 102 and then routing those requests (shown here as 110, 111, 112) to a server selected according to policies implemented in the load-balancing host software. Note that the requests 110, 111, 112 are shown being sent to specific Web servers: this is merely an example of a possible outcome of the load balancing process. Load-balancing techniques are known in the art, and do not form part of the present invention. (Column 8, lines 49 to 58.) Load balancing happens on an arbitrary basis as far as sessions are concerned which of course is not significant because the whole point of Bayeh is that irrespective of which server receives a request, the session object can be made available.

The nature of the servlets, and the fact that they are not assigned to any particular session is underlined when one considers subsequent parts of Bayeh:

If a servlet is written not to use session-related information, the servlet will perform its specific processing, eventually finishing and returning its results to the server through the response object with which it was invoked. While the servlet may have been invoked with a valid session ID present in the request object, it will not have made use of that session. (Column 10, lines 54 to 60.)

and

If session services are required for this servlet's application, then the session services features of the present invention are used. According to the present invention, a servlet in this scenario will include code to get the session object for this session, and update that object to reflect session information related to the servlet's processing. When the servlet processing is finished, the session object is returned to the session pool, where it can be accessed for subsequent transactions with this servlet or a different servlet in the clustered environment. In this way, the state of the session can be communicated among the clustered servers and their servlets. (Column 10, line 64 to column 11, line 8.)

The features of Applicant's invention as recited in claim 1 are different from Bayeh, because in Applicant's invention The sessions are between a plurality of terminals and a server. The server has a plurality of threads. The sessions are grouped into a plurality of groups and a "thread" is assigned to each "group"

of sessions. Each assigned thread only handles the events of the group of sessions. This is not disclosed or suggested by Bayeh. Thus claim 1 cannot be anticipated by Bayeh. Independent claims 18, 21, and 22 contain features corresponding to the features of claim 1 and likewise are not anticipated. Claims 2-17 and 19-21 should be allowable at least be reason of their respective dependencies.

The Applicant would like to respond to the Examiner's assertions that threads are servlets. The Applicant disagrees with this view. However, irrespective of this difference of opinion, the Applicant has demonstrated in the preceding argumentation that the claims are not anticipated by Bayeh.

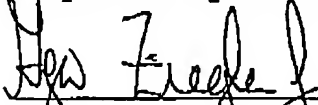
3. Claims 5-8, 13, 14, and 17 are not unpatentable over Bayeh under 35 U.S.C. §103(a) for at least the above-stated reasons. Additionally, Bayeh is concerned with a very different problem, than that which is addressed by the features of claim 1, which is making available the appropriate session object in response to a request which forms part of a session in a clustered server environment in which requests are not tied to particular servers. Therefore, while not only does Bayeh not contain any teaching that would lead to the invention of claim 1, Bayeh does not provide any motivation for modification, because it would be to achieve something which is irrelevant to the way Bayeh works. Thus, a *prima facie* case of obviousness cannot be established.

For all of the foregoing reasons, it is respectfully submitted that all of the claims now present in the application are clearly novel and patentable over the prior art of record, and are in proper form for allowance. Accordingly, favorable reconsideration and allowance is respectfully requested. Should any unresolved issues remain, the Examiner is invited to call

Applicants' attorney at the telephone number indicated below.
Entry of this amendment is solicited for purposes of appeal.

The Commissioner is hereby authorized to charge payment of a three-month extension of time and any other fees associated with this communication or credit any over payment to Deposit Account No. 16-1350.

Respectfully submitted,



Geza C. Ziegler, Jr.
Reg. No. 44,004

5 December 2005
Date

Perman & Green, LLP
425 Post Road
Fairfield, CT 06824
(203) 259-1800
Customer No.: 2512

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being transmitted by facsimile to (571) 273-8300 on the date indicated below, addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date: Dec 5, 2005

Signature: Meaghan Baye

Meaghan Baye
Person Making Deposit